CD/NSWC MSSPO (102) CR-96/02

INTEGRATION OF A CONSTITUTIVE EQUATION BASED FINITE ELEMENT FOR COMPOSITE MATERIALS INTO A COMMERCIAL FINITE ELEMENT SOFTWARE CODE

FINAL REPORT
SwRI Project No. 06-7171

Prepared by:
Mark Jones

Southwest Research Institute

April 1996

Sponsoring Organization:
The Advanced Research Projects Agency

SOUTHWEST RESEARCH INSTITUTE
SAN ANTONIO HOUSTON
DETROIT WASHINGTON, DC

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) SwRI Project Number 06-7171	5. MONITORING ORGANIZATION REPORT NUMBER(S) CD/NSWC MSSPO (102) CR-96/02
6a. NAME OF PERFORMING ORGANIZATION Southwest Research Institute 6b. OFFICE SYMBOL *(if applicable)*	7a. NAME OF MONITORING ORGANIZATION Carderock Division Naval Surface Warfare Center
6c. ADDRESS *(City, State, and ZIP Code)* 6220 Culebra Road San Antonio, TX 78228-0510	7b. ADDRESS *(City, State, and ZIP Code)* Bethesda, MD 20084-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Advanced Research Projects Agency 8b OFFICE SYMBOL *(if applicable)* TTO	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00167-95-C-0064
8c. ADDRESS *(City, State, and ZIP Code)* 3701 North Fairfax Drive Arlington, VA 22203-1714	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO.: 064705D PROJECT NO.: TASK NO.: WORK UNIT ACCESSION NO.: 95-1-1020-521

11. TITLE *(Include Security Classification)*

Integration of a Constitutive Equation Based Finite Element for Composite Materials Into a Commerical Finite Element Software Code (U)

12. PERSONAL AUTHOR(S)

Mark Jones

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Year, Month, Day)*	15. PAGE COUNT
Contractor Report	FROM 4/95 TO 4/96	4/5/96	16

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
FIELD	GROUP	SUB-GROUP	
			composites, thick laminate composites, finite element

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This report discusses the integration of a novel finite element for analyzing composite material structures into a commercially available finite element software package. This element is a constitutive equation based formulation that predicts individual layer strains and stresses in a computationally efficient manner. Implementing this element into a commercial code will improve its efficiency for predicting the structural response of composite material structures.

The work discussed in this report involved utilizing algorithms previously developed at Southwest Research Institute (SwRI) that are based on the constitutive equations and integrating them with the ANSYS finite element code. An example problem using this method is presented that predicts the laminate free edge stress distribution that is important for examining potential delaminations at design features such as holes, cutouts, and joints.

The results of this study show that the element is accurate for determining the stress distribution at the laminate free edge. Recommendations for enhancements to the element are made to improve its modeling efficiency even further.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL Keith R. McAllister	22b. TELEPHONE *(Include Area Code)* (301) 227-1672	22c. OFFICE SYMBOL CD/NSWC MSSPO (Code 102)

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

CD/NSWC MSSPG (102) CR-96/02

INTEGRATION OF A CONSTITUTIVE EQUATION BASED FINITE ELEMENT FOR COMPOSITE MATERIALS INTO A COMMERCIAL FINITE ELEMENT SOFTWARE CODE

FINAL REPORT
SwRI Project No. 06-7171

Prepared by:
Mark Jones

Southwest Research Institute
6220 Culebra Road
San Antonio Texas, 78228-0510

April 1996

Contract Number N00167-95-C-0064

Sponsoring Organization:
The Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington, Virginia 22203-1741

Reviewed By:

Edward M. Patton, Ph.D., Manager
Marine Engineering Section

Approved By:

Edward M. Briggs, Director
Marine Technology Department

The Contractor, Southwest Research Institute, hereby certifies that, to the best of its knowledge and belief, the technical data delivered herewith under Contract No. N00167-95-C-0064 is complete, accurate, and complies with all requirements of the contract.

April 3, 1996
Date

Geoffrey Dearnaley, Vice President
Materials and Structures Division

EXECUTIVE SUMMARY

This report discusses the integration of a novel finite element for analyzing composite material structures into a commercially available finite element software package. This element is a constitutive equation based formulation that predicts individual layer strains and stresses in a computationally efficient manner. Implementing this element into a commercial code will improve its efficiency for predicting the structural response of composite material structures.

The work discussed in this report involved utilizing algorithms previously developed at Southwest Research Institute (SwRI) that are based on the constitutive equations and integrating them with the ANSYS finite element code. An example problem using this method is presented that predicts the laminate free edge stress distribution that is important for examining potential delaminations at design features such as holes, cutouts, and joints.

The results of this study show that the element is accurate for determining the stress distribution at the laminate free edge. Recommendations for enhancements to the element are made to improve its modeling efficiency even further.

TABLE OF CONTENTS

		Page
EXECUTIVE SUMMARY		ii
LIST OF FIGURES		iv
LIST OF TABLES		iv
1.0	INTRODUCTION	1
2.0	BACKGROUND	2
3.0	TECHNICAL APPROACH	4
4.0	RESULTS	5
4.1	Model Description	5
4.2	Results	9
5.0	CONCLUSIONS	14
6.0	RECOMMENDATIONS	15
7.0	REFERENCES	16

APPENDIX A Fortran Source Code Listing

APPENDIX B Example Problem Results

APPENDIX C Diskette Containing Source Code Listing

LIST OF FIGURES

Page

FIGURE 1 Ply Orientation of Plate 6

FIGURE 2 Finite Element Model 7

FIGURE 3 Displacements in X-Direction 10

FIGURE 4 Physical Interpretation of Symmetry Plane 11

FIGURE 5 Displacements in Y-Direction 12

FIGURE 6 Stress Across Plate Width 13

LIST OF TABLES

Page

TABLE 1 Material Properties 5

1.0 INTRODUCTION

This report discusses the integration of a novel finite element for analyzing composite material structures into a commercial finite element software code. The work described within was Task 3.0 of the U.S. Government contract N00167-95-C-0064, and is intended as a stand-alone document. The government report numbered CD/NSWC MSSPO (102) CR-95/02 and entitled "Development of Large, Thick Laminate Structures for Offshore Applications", discusses the rest of the tasks associated with this program.

The thrust of this effort was to develop a computationally effective method to analyze the through-thickness stress and strain distribution of thick composite parts as applied to offshore structures. However this method can used on a variety of composite material applications for structures.

This report is organized into the following sections: background information and the rationale for using this method are presented in Section 2.0. Section 3.0 discusses the technical approach for integrating the constitutive based model into a commercial finite element program. The results of an example problem based on the free edge stress distribution problem are presented in Section 4.0. Section 5.0 discusses the conclusions of using this method. Recommendations for future efforts are discussed in Section 6.0. Appendix A presents the source code listing for integrating the model into the commercial finite element code. Results from the example problem are presented in Appendix B.

2.0 BACKGROUND

The application of composite structures for offshore structures will require parts that are thick (relative to typical aerospace structures), and have loads that are applied through the thickness plane of the structure. Particular attention must be paid to potential delamination of the composite material near design details such as holes, cutouts, and both composite-to-composite and steel-to-composite joint regions.

Therefore an analytical method that can determine the structural response of these laminated parts must be used to ensure their soundness when placed into service. Due to the complex geometry and loading conditions acting on this parts, the finite element method must be employed to adequately analyze their structural behavior. Moreover, the method used must be computationally efficient to permit analyzing these structures in a timely and cost-effective manner.

There are many methods available to describe the mechanical response of layered composite materials. Typically these methods are based on classical laminated plate theory with estimates of the bulk mechanical response for thick composite sections. Classical laminate plate theory is very effective in modeling the response of thin laminates when there are no significant loads applied in the thickness direction of the laminate. However, when this is not the case, some other material description must be used to determine the through-thickness mechanical response.

One common approach to describe the through thickness material properties is to replace the layer by layer characteristics with an average set of orthotropic constitutive properties that are used to model many layers of composite. This method is adequate for large sections of composite, but fail to determine the layer-by-layer behavior that is critical information near design details. One means to get around this problem is to employ "Global-Local" modeling, where a very refined layer-by-layer finite model mesh is used around the local design details, and a coarser mesh with effective properties is used for the bulk response of the structure. Two major problems with this method are that the analyst must know beforehand what design features are important for the bulk mechanical response of the structure, and if there are many design features of interest then the refined layer-by-layer mesh portions of the model can grow to the point where computation becomes prohibitively expensive. It is clear that what is needed is a reliable three dimensional material description that can be implemented into a finite element that captures the essential features of a layered composite without the computational expense of a layer-by-layer approach.

There are many methods proposed in the literature for the description of the three dimensional "effective" properties of a laminated composite. However, these methods generally have very complex mathematical descriptions of these properties and require extensive numerical matrix manipulations. These characteristics result in finite element solutions that are computationally prohibitive for modeling composite structures of any great size or detail.

A method that overcomes these obstacles is a generalized averaging procedure for the description of the mechanical properties of a layered composite that was proposed by

Christensen [1]. The main point of this procedure is the separation of the fiber-dominated properties from the matrix dominated properties. The matrix dominated properties are described by two independent constants that are the result of this averaging procedure, and are used in a manner that is similar to Hooke's Law. The fiber dominated properties are then averaged into one constant that the provides an additive term to the constitutive formulation in the direction of the fibers in that layer of the composite.

This constitutive equation method was adapted into a displacement-based three dimensional finite element code by Patton [2]. This code, which will be referred to as the SwRI code, is a "university-grade" code, meaning it does not have a pre- and post processor and has limited functionality. In order to utilize this method in a cost-effective manner, it is necessary to implement it into a commercial grade FEA code.

3.0 TECHNICAL APPROACH

The technical approach of this effort was to use portions of the SwRI code specific to the single element formulation and integrate it with a commercial finite element program. There are several factors that must be considered before deciding on which software package to use.

The most important characteristic for choosing which finite element code to integrate the constitutive equation based element is that the commercial code have the capability to accept user written elements. Also the present formulation of the constitutive equation based element requires that the finite element program have a displacement based wavefront solver. This is the most common means of solving finite element models, so there are a large number of codes available. Another requirement is that the program preferably have an integrated pre- and post processor, or at least be able to interface with a third-party pre- and post processing program such as PATRAN or IDEAS. A highly desirable requirement is that the program is already in place within the organization and has an experienced user base.

There are a number of finite element codes that meet these requirements. NASTRAN, ANSYS, and ABAQUS among the better known products. There is no one program that is the best choice, the decision being based on cost, familiarity, and of course whether or not if one already possess one of these codes. The best choice for the Marine Technology Department at SwRI was ANSYS 5.1 coded to run a HP 700 series UNIX Workstation. It should be noted that the user element interface capability is not available on the PC-DOS version of this software package.

An interface to the ANSYS code is provided by the use of User Programmable Features (UPFs) [3]. UPFs are a set of FORTRAN routines that allow the user to write custom elements, loading conditions, material properties, and other user specific applications.

This particular application required the use of the user element subroutines. These subroutines provide the interface between the user element and the ANSYS code. The user is responsible for developing the entire element formulation including the load vector, stiffness matrix, and material matrix. The user must also address issues with ANSYS variable passing, input/output switches, and any options for the element.

The portions specific to single element formulations in the SwRI code were used in the ANSYS user routine. Because the SwRI code was developed on a DOS based PC with Microsoft FORTRAN V5, some of the code had to be changed to successfully port it to the HP Workstation. The source code listing for this element is presented in Appendix A.

4.0 RESULTS

An example problem was tested to determine the accuracy and utility of implementing the constitutive formulation with ANSYS. The geometry, material properties and boundary conditions for the example problem were identical to the one solved by Pipes and Pagano [4]. This problem is a free edge stress problem that consists of a 4-ply laminate under uniform axial extension. This is one of the most demanding solutions for laminate analyses, in that there are interlaminar stresses at the free edge of the laminate that are not accounted for by classical laminate theory. Additionally, this same problem was solved by the existing SwRI code and compared to the results from the ANSYS model.

4.1 Model Description

The geometry and corresponding finite element mesh was created in the interactive "mode" of ANSYS, which is a graphical interface. This model is a flat plate that is 1.2 inches long, by 0.8 inches wide by 0.2 inches thick. As shown in Figure 1, it is a +45/-45/-45/+45 composite structure. The finite element model consists of 96 8-noded brick elements and is shown in Figure 2. There are two symmetry planes, one along the x-axis, and one along the y-axis. The loading is a uni-axial extension applied by displacements along one surface of the plate in the x-direction. The value of these displacements was 0.075 inch at each node of the surface.

The material properties for this problem are shown in Table 1. These values are input using the ANSYS MP command via the command line.

TABLE 1- MATERIAL PROPERTIES

Material Property	ANSYS Variable	Value
Young's Modulus along fiber direction	EX	2.1×10^7 psi
Young's Modulus perpendicular to fiber direction	EY	2.0×10^6 psi
Possions Ratio	NUXY	0.21
Shear Modulus in xy plane	GXY	0.85×10^6 psi
Shear Modulus in xz plane	GXZ	0.0
Density	RHO	0.0
Coefficient of thermal expansion in x direction	ALPHAX	0.0
Coefficient of thermal expansion in y direction	ALPHAY	0.0
Coefficient of thermal expansion in z direction	ALPHAZ	0.0

The individual ply layer properties are input using ANSYS real constants. It was found that the parameters for the layers must be entered with two material properties through the lamina

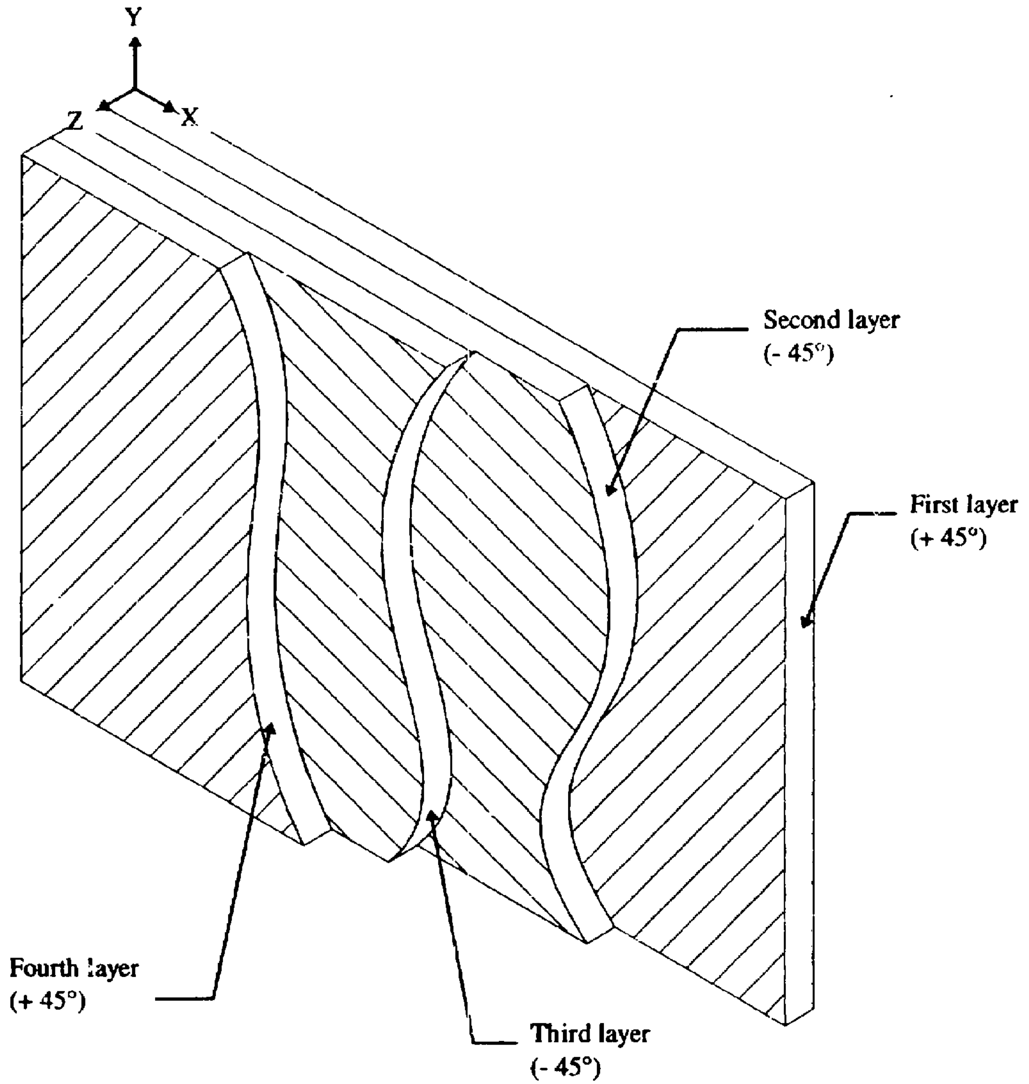

FIGURE 1 - Ply Orientation of Plate

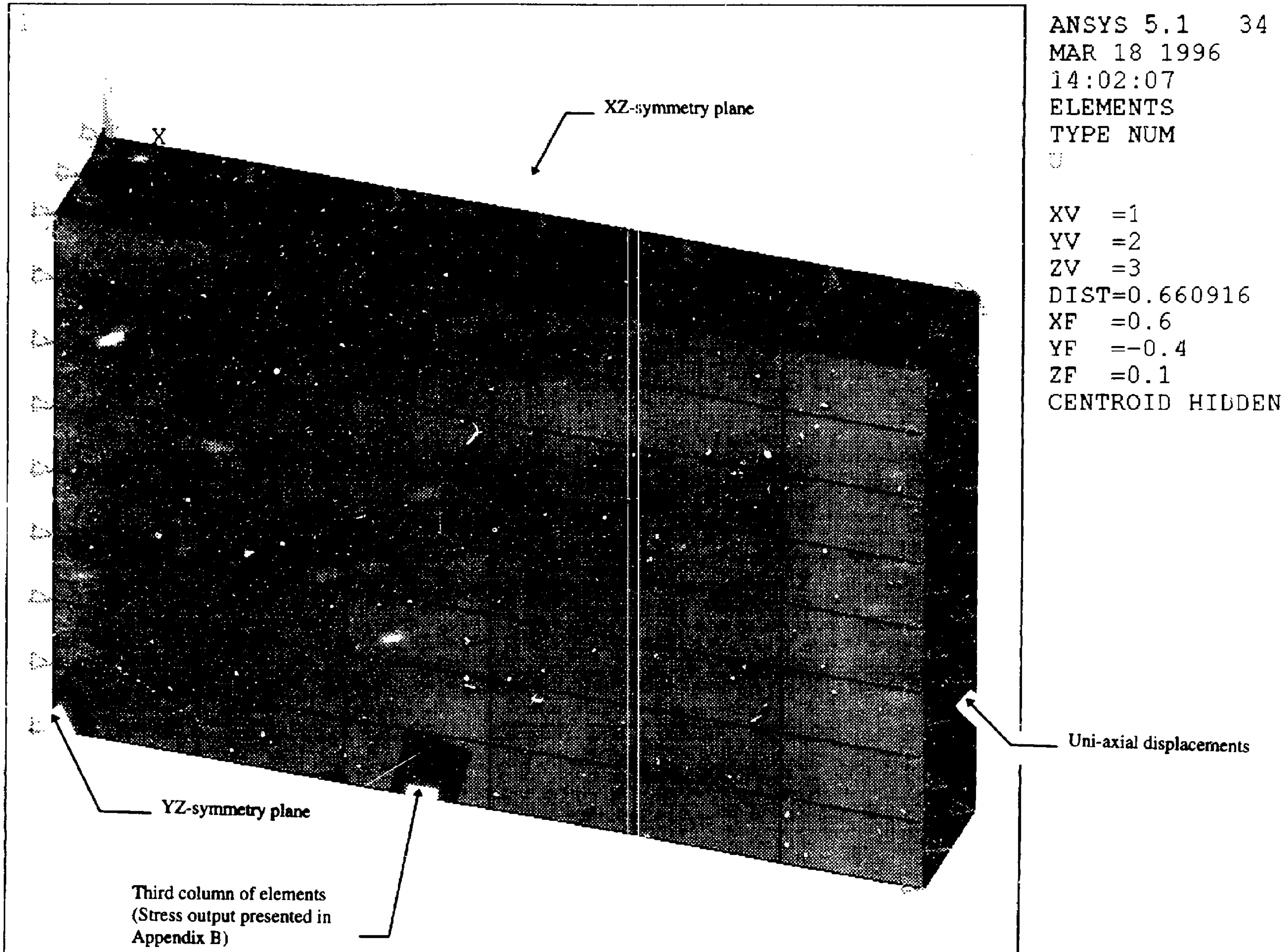

FIGURE 2 - Finite Element Model

thickness. This requires that there are two elements through the thickness, each with a different ANSYS material property ID number. However, the material properties are identical for both material sets. It is suspected that this condition is caused by the formulation of the material matrix in the SwRI code and will eventually need to be rectified. Therefore a separate real constant is required for each set of elements through the laminate thickness.

The real constant data consists of the number of layers through the element, layer material numbers, fiber orientations, and layer thickness. This data is entered using the ANSYS commands R and RMORE via the command line. The format for entering the real constants is based on an existing ANSYS composite element, SOLID 46, and is shown below:

R,*N1,NL,R2,R3,R4,R5,R6*

Where: N1 is the real constant ID set
NL is the number of layers in this set
R2 - R6 are not used

RMORE,*R7,R8,R9,R10,R11,R12*

Where: R7 - R12 are not used

RMORE,*R13,R14,R15,R16,R17,R18*

Where: R13 is the material ID for the first layer in this set
R14 is the fiber direction from element x-axis for first layer in this set
R15 is the lamina thickness of the first layer in this set
R16 is the material ID for the second layer in this set
R17 is the fiber direction from element x-axis for second layer in this set
R18 is the lamina thickness of the second layer in this set

The command line data entry for this problem is as follows:

(First real constant set, layers 1-2)
```
R,1,2,,,,,
RMORE,,,,,
RMORE,1,45,0.05,1,-45,0.05
```

(Second real constant set, layers 3-4)
```
R,2,2,,,,,
RMORE,,,,,
RMORE,2,-45,0.05,2,45,0.05
```

4.2 Results

The contour plot for the direction of the axial displacements is shown in Figure 3. The first item to note is how the displacements behave at the xz-symmetry plane. Instead of acting like the adjoining row of elements, the displacements increase at the symmetry plane. This behavior should not happen at a symmetry plane, and can be explained by the fact that a symmetry plane implies isotropy, which is not the case for lamina with fiber orientations. The fiber angles for each layer are mirrored by the boundary conditions as shown in Figure 4, which results in a discontinuity. At this boundary the fiber angles are reversed which is why the displacements are reversed in reference to the free edge at the other side of the plate.

The second feature of the displacement results show difference in displacements between the +45° and the -45° layers through the plate thickness. This can be seen in Figure 3, and Figure 5 which shows the contour plot for displacements perpendicular to the load direction. This is expected behavior for the given laminate ply orientations.

The stress output was not incorporated into the ANSYS post-processor due to time constraints. The stress values for each layer is available in tabular form, and is presented in Appendix B for the third column of elements as shown in Figure 2. A graph of the stresses across the plate width for the second lamina layer is shown in Figure 6.

The results of Pipes and Pagano revealed that at the free edge of a laminate the interlaminar stresses (τ_{xz}) grow, while the in-plane shear stresses (τ_{xy}) diminish to zero. Also there is a decrease in the stresses acting along the load direction (σ_x) at the free edge. The results shown in Figure 6 follow this behavior, but with some discrepancies. The first discrepancy takes place from the xz-symmetry plane to y/b=0.3. Here the stresses are effected by the presence of the symmetry plane and can be discounted. The second discrepancy occurs near the free edge, in that τ_{xy} does not diminish to zero, and that τ_{xz} does not increase as much as predicted by Pipes and Pagano. This phenomena can be explained by the fact that an 8-noded element has a linear displacement distribution and can not fully capture the rapidly changing structural response at the free edge. Two methods to capture this response is to use a finer mesh at the free edge and/or use 20-noded brick elements.

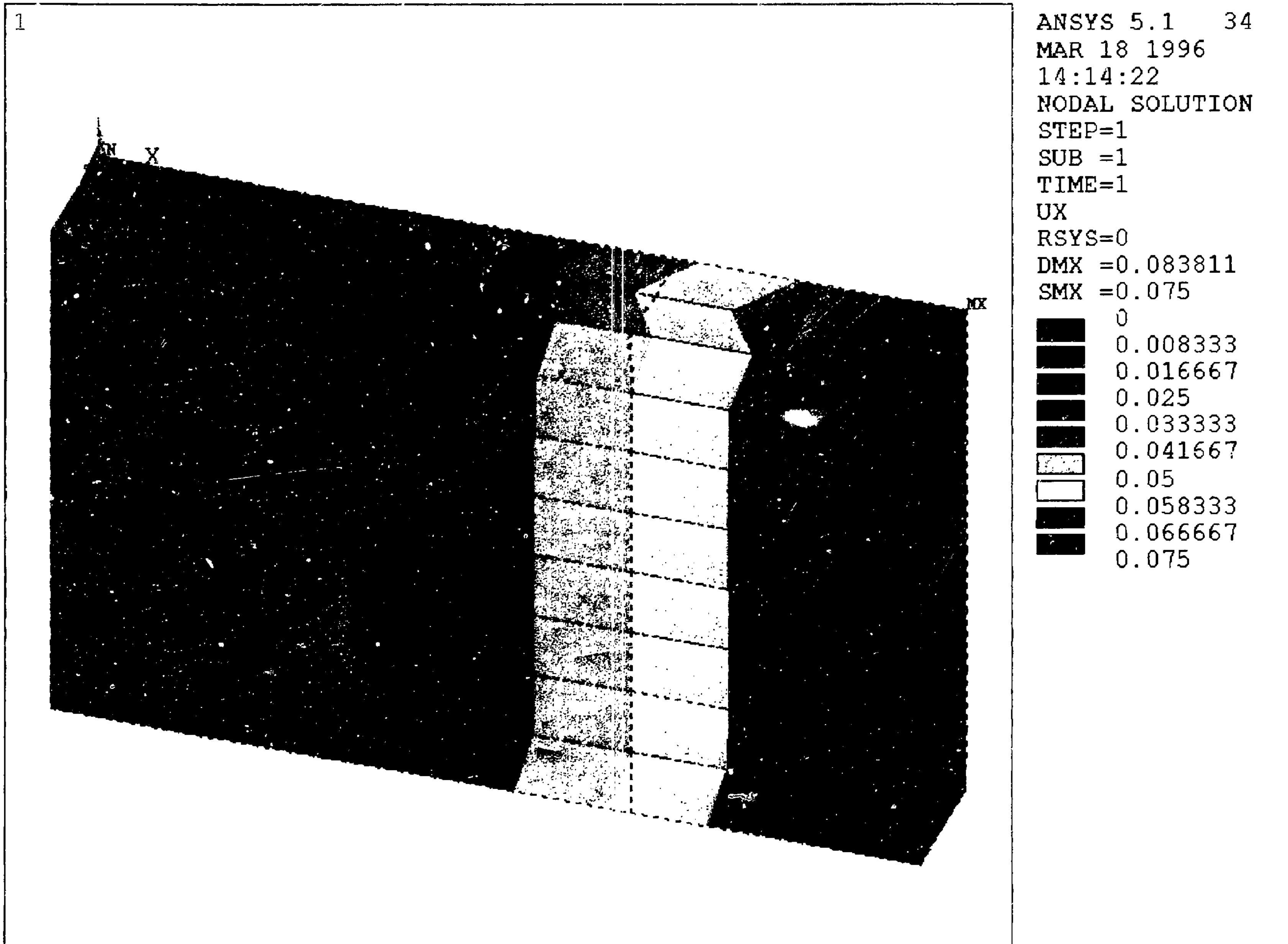

FIGURE 3 - Displacements in X-Direction

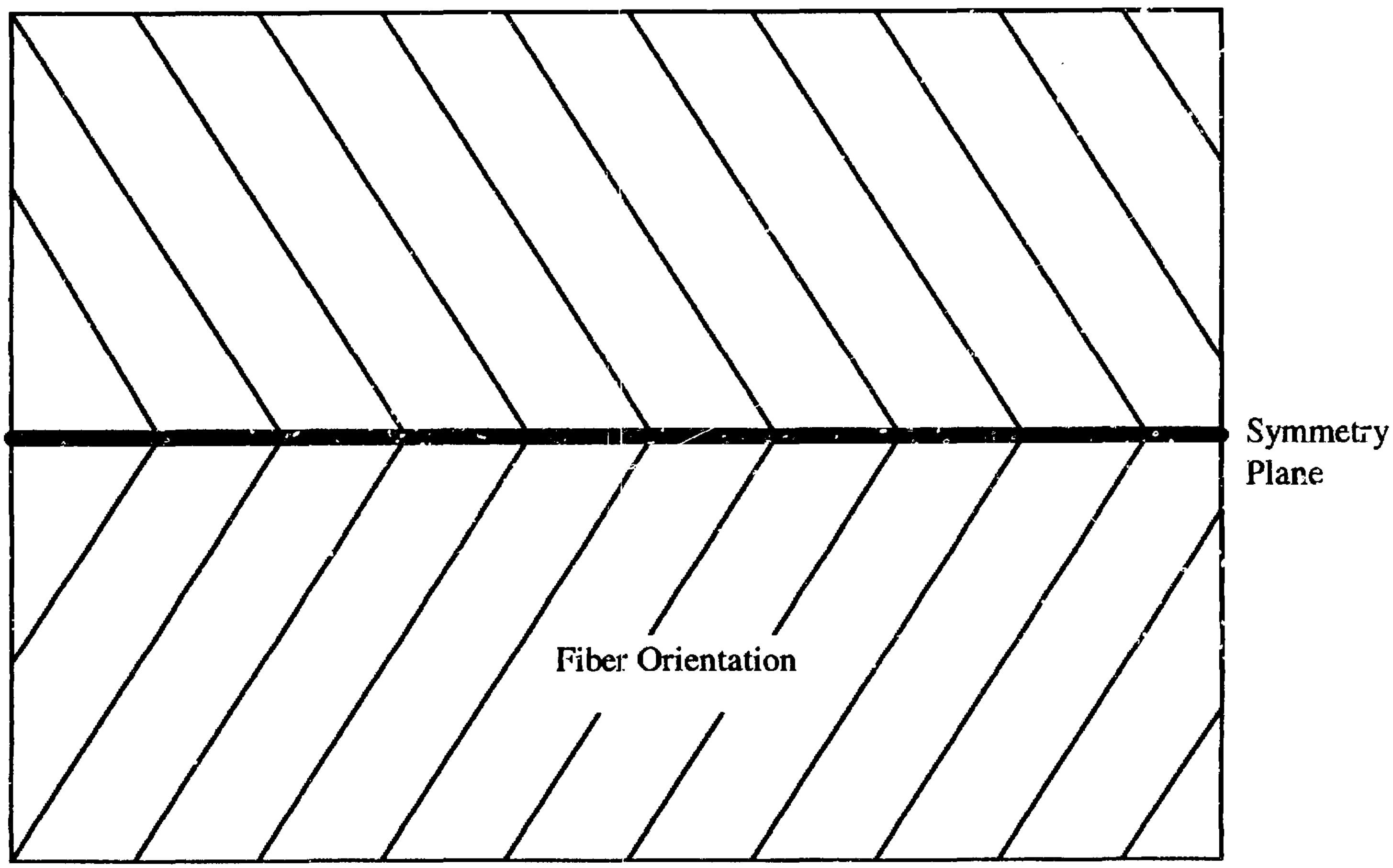

FIGURE 4- Physical Interperation of Symmetry Plane

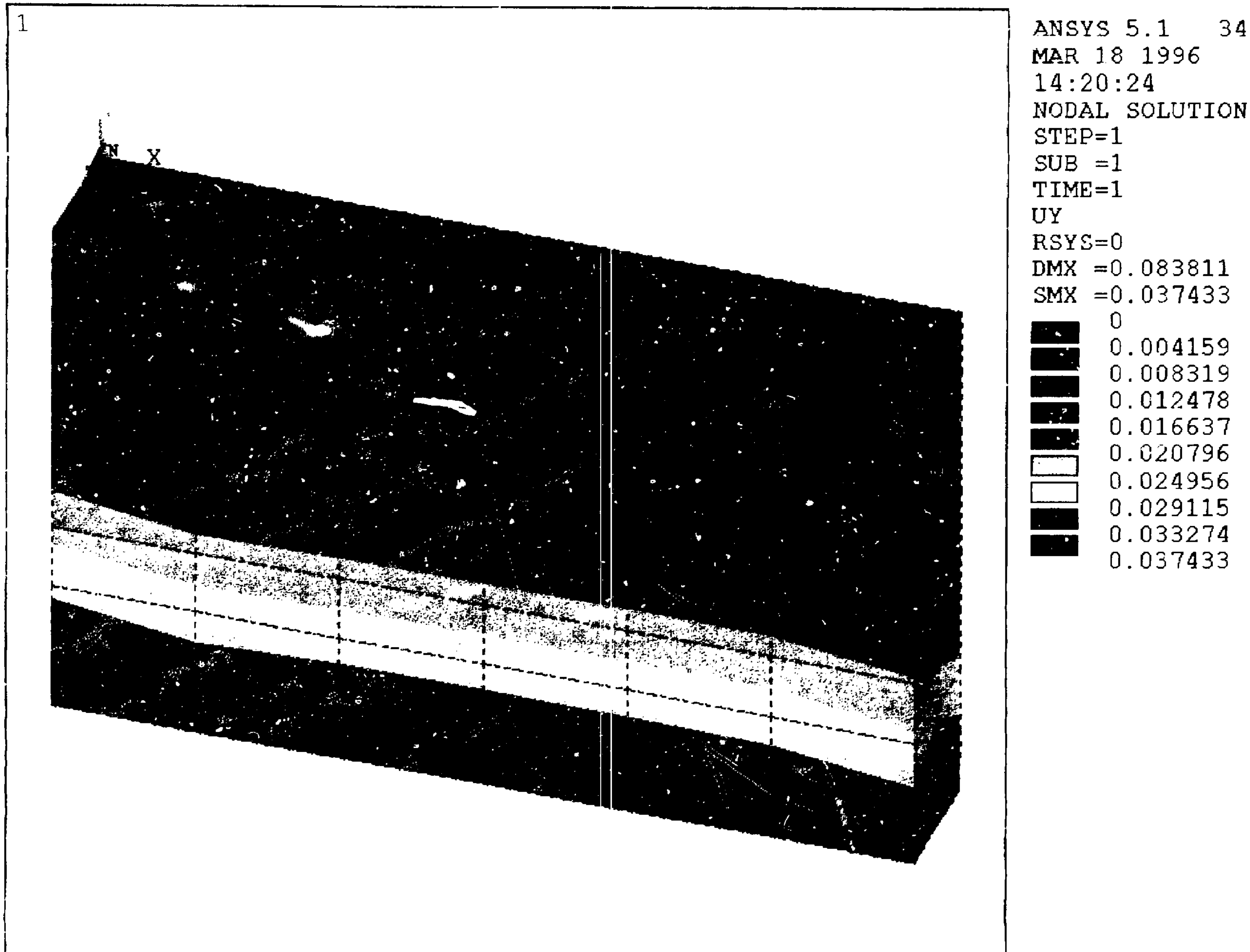

FIGURE 5 - Displacements in Y-Direction

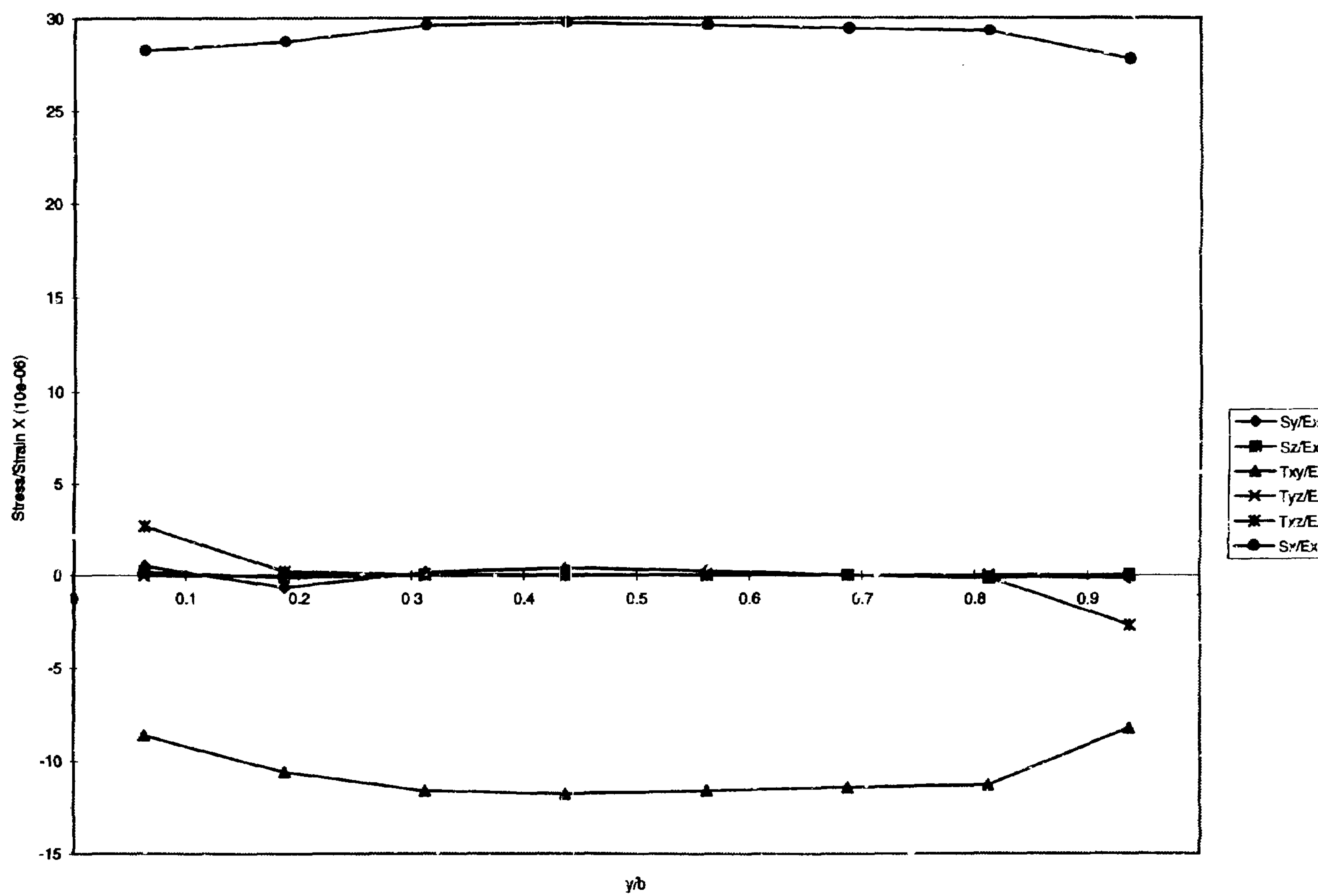

FIGURE 6 - Stress Across Plate Width

5.0 CONCLUSIONS

The Christensen constitutive equation based finite element method has successfully been integrated into ANSYS 5.1. The results follow the stress distribution at the laminate free edge according to the results of Pipes and Pagano. Therefore with proper modeling the element has the accuracy for examining potential delaminations at design features. Proper modeling techniques include not using symmetry planes and using a fine mesh at the laminate free edge.

The problem with using symmetry planes is that it implies isotropy, which was shown not to be the case for the element formulation. For this reason, symmetry planes are not recommended for modeling. Further development of the element formulation algorithms is needed to take advantage of symmetry planes.

The element formulation requires that there are two material sets through the lamina thickness. Using only one material set gives inaccurate results. Again further development of the element formulation algoritims are needed to correct this deficiency.

The present element formulation is an 8-noded isoparametric brick. To accurately capture the rapidly changing stress field near the laminate free edge, the mesh will have to be finer that the rest of model. Another way to better capture this stress field is to incorporate a 20-noded brick element with the ANSYS code.

Writing software code that interfaces with an existing program always presents challenges and results in a product that is not as tightly coupled as desired. The greatest difficulty in writing the user element was dealing with the input/output with ANSYS. Determining how to bring in variables from ANSYS and storing results was not intuitive and poorly documented. ANSYS offers a short course on writing user programmable features and in retrospect this would have been a valuable course for developing this element UPF.

6.0 RECOMMENDATIONS

The efficiency of using this element could be improved by more development to the algorithms and tighter integration with ANSYS. Specific enhancements are allowing the user to use the graphical input for material properties and real constants, adding algorithms to take advantage of symmetry planes, refining the algorithms so that only one material property is needed, incorporating 20-node brick elements, and writing the code to the view the strain and stress output in graphical form.

The best way to implement these enhancements would be to improve the element formulations and incorporate them into the existing ANSYS composite element SOLID 46. This approach would give the tightest coupling between the Christensen formulation and would result in the most efficient use of this element.

7.0 REFERENCES

1. Christensen, R.M., and Zywicz, E., "A Three Dimensional Constitutive Theory for Fiber Composite Laminates." ASME Paper 90-WA/APM-32, presented at the Winter Annual Meeting of the ASME, Nov. 25-30, 1990, Dallas, TX.

2. Patton, E.M., "Implementation of an Efficient Description of Elastic Properties of a Layered Composite in a Finite Element," Computers in Engineering, Vol 2, pp 91-94, ASME 1992.

3. Swanson Analysis Systems, Inc., "ANSYS Programmer's Manual for Revision 5.0," Houston, PA, March 15, 1993.

4. Pipes, R.B., and Pagano, N.J., "Interlaminar Stresses in Composite Laminates Under Uniform Axial Extension," Journal of Composite Materials, V.4, pp.538-548, 1970.

APPENDIX A

```
*DECK UEC102    PARALLEL            USER     2/23/90             PCK
      SUBROUTINE UEC102 (ELCDN,IELC,KERR)
C     *** THIS SUBROUTINE DEFINES THE CHARACTERISTICS OF USER102.
C     **** SEE UEC100 FOR DESCRIPTIVE COMMENTS **
C
      INCLUDE 'IMPCOM'
      INCLUDE 'ECHPRM'
C
      EXTERNAL  NMINFO,ALTINF,ANSERR,ERINQR,WRINQR
      INTEGER   ERINQR,WRINQR,IOTT
      INTEGER IELC(*),I,KERR,KY2
      CHARACTER*28  ELCDN
C
C        *** ANSYS(R) COPYRIGHT(C) 1971,78,82,83,85,87,89,92
C        *** SWANSON ANALYSIS SYSTEMS, INC.
C
C     **** DEFINE ELEMENT NAME ***
      CALL NMINFO (IELC(1),'USER102 ')
      CALL ALTINF (IELC(1),'        ')
      ELCDN = 'USER ELEMENT 102           '
C
C     **** ELEMENT TYPE CHARACTERISTICS:
C
C     **** BASED ON SOLID46 ELEMENT
C
      IELC(KDIM  ) = 3
      IELC(ISHAP ) = JBRICK
      IELC(IDEGEN) = 3
      IELC(MSHLIM) = 1
      IELC(KELSTO) = 12
      IELC(KDOFS ) = UX + UY + UZ
      KY2 = IELC(KYOP2 )
C
      IF (KY2 .LE. 1) THEN
        IELC(NMLAYS) = -1
        IF (KY2 .EQ. 1) IELC(NMLAYS) = -2
        IELC(LCANGL) = 7
        IELC(NMDRLC) = 312
        IELC(MATRQD) = 1
        IELC(MATMUL) = 1
      ELSE
        IELC(NMLAYS) = -3
        IELC(LCANGL) = 128
        IELC(NMDRLC) = 128
      ENDIF
      IELC(NMNDMX) = 8
      IELC(MATRXS) = STIFM + MASSM + SSTIFM
      IELC(NMPRES) = 6
      IELC(NMTEMP) = 8
      RETURN
      END
```

```
      SUBROUTINE UEL102 (ELEM,IELC,ELMDAT,EOMASK,NODES,LOCSVR,KELREQ,
     X KELFIL,NR,XYZ,U,KELOUT,ZS,ZASS,DAMP,GSTIF,ZSC,ZSCNR,ELVOL,ELMASS,
     X CENTER,ELENER,EDINDX,LCERST)
C --- SEE EL100 AND EL101 FOR DOCUMENTATION ---
C
C
C      *** ANSYS(R) COPYRIGHT(C) 1971,78,82,83,85,87,89,92
C      *** SWANSON ANALYSIS SYSTEMS, INC.
C
C INPUT ARGUMENTS:
C     ELEM   (INT,SC,IN)        - ELEMENT LABEL (NUMBER)
C     IELC   (INT,AR(IELCSZ),IN) - ARRAY OF ELEMENT TYPE CHARACTERISTICS
C     ELMDAT (INT,AR(10),IN)    - ARRAY OF ELEMENT DATA
C     EOMASK (INT,SC,IN)        - BIT PATTERN FOR ELEMENT OUTPUT
C                                 (SEE OUTPCM)
C     NODES  (INT,AR(NNOD),IN)  - ARRAY OF ELEMENT NODE NUMBERS
C     LOCSVR (INT,SC,IN)        - LOCATION OF THE SAVED VARIABLES
C                                 ON FILE .ESAV FOR THIS ELEMENT
C     KELREQ (INT,AR(10),IN)    - MATRIX AND LOAD VECTOR FORM REQUESTS
C                                 (INDICES FOR KELREQ ARE GIVEN WITH OUTPUT
C                                 ARGUMENTS BELOW)
C     KELFIL (INT,AR(10),IN)    - KEYS INDICATING INCOMING MATRICES AND
C                                 LOAD VECTORS (INDICES FOR KELFIL ARE THE
C                                 SAME AS GIVEN FOR KELREQ WITH OUTPUT
C                                 ARGUMENTS BELOW)
C     NR     (INT,SC,IN)        - MATRIX AND LOAD VECTOR SIZE
C     XYZ    (DP,AR(6,NNOD),IN) - NODAL COORDINATES (ORIG) AND ROTATION ANGLE
C     U      (DP,AR(NR,5),IN)   - ELEMENT NODAL SOLUTION VALUES
C
C OUTPUT ARGUMENTS:
C     KELOUT (INT,AR(10),OUT)   - KEYS INDICATING CREATED MATRICES AND
C                                 LOAD VECTORS (INDICES FOR KELOUT
C                                 ARE THE SAME AS FOR KELREQ BELOW)
C     ZS     (DP,AR(NR,NR),INOUT)- K MATRIX         (KELREQ(1))
C     ZASS   (DP,AR(NR,NR),INOUT)- M MATRIX         (KELREQ(2))
C     DAMP   (DP,AR(NR,NR),INOUT)- C MATRIX         (KELREQ(3))
C     GSTIF  (DP,AR(NR,NR),INOUT)- S MATRIX         (KELREQ(4))
C     ZSC    (DP,AR(NR),OUT)    - APPLIED F VECTOR  (KELREQ(5))
C     ZSCNR  (DP,AR(NR),OUT)    - N-R RESTORING F VECTOR (KELREQ(6))
C                                 OR IMAGINARY F VECTOR (KELREQ(7))
C     ELVOL  (DP,SC,OUT         - ELEMENT VOLUME
C     ELMASS (DP,SC,OUT)        - ELEMENT MASS
C     CENTER (DP,AR(3),OUT)     - CENTROID LOCATION
C     ELENER (DP,AR(5),OUT)     - ELMENT ENERGIES
C     EDINDX (INT,AR(20),OUT)   - ELEMENT RESULT DATA FILE INDEXES
C     LCERST (INT,SC,INOUT)     - POSITION ON RESULT FILE
C
C   ---- START OF COMDECKS
C
      INCLUDE 'IMPCOM'
      INCLUDE 'ECHPRM'
      INCLUDE 'ELPARM'
      INCLUDE 'ELUCOM'
C
```

```
C    ---- END OF COMDECKS
C
     EXTERNAL TRACK,ANSERR,SVGIDX,SVRGET,RVRGET,WRINQR,PROPE1,
    X VZERO
C
C    --- ARGUMENTS
     INTEGER ELEM,IELC(IELCSZ),ELMDAT(10),EOMASK,NODES(8),LOCSVR,
    X KELREQ(10),KELFIL(10),NR,KELOUT(10),EDINDX(20),LCERST,WRINQR
C
C    --- ARGUMENTS
     DOUBLE PRECISION
    X XYZ(6,8),U(NR,5),ZS(NR,NR),ZASS(NR,NR),DAMP(NR,NR),GSTIF(NR,NR),
    X ZSC(NR),ZSCNR(NR),ELVOL,ELMASS,CENTER(3),ELENER(5)
C
C    --- LOCAL
C
     INTEGER
    X SVINDX(10),
    X PMAT,IREAL,PREAL,NRVR,NSSVR,I,J,IOTT,
    X K,KY2,NUMLAYERS,KP1,KP2
C
C
C
C    --- PROPERTIES
C
     DOUBLE PRECISION
    X PROP(10),PROPO,
    X THK(100)
C
C    --- RVR,SSVR
C
     DOUBLE PRECISION
    X RVR(312), SSVR(10),SVIDNX(1),TK
C
     INTEGER ANGFIB(100,100),MATNUM(250),THETA
     DOUBLE PRECISION PROPS(10)
C
CALL TRACK (5,'UEL102')
C
C    --- DEFINE INITIAL DATA
C
     MATNUM(ELEM) = ELMDAT(PMAT)
IREAL = ELMDAT(PREAL)
C
C    --- IELC POINTERS DEFINED IN ECHPRM AND ELCCMT
C
     NRVR = IELC(NMTRLC)
     NSSVR = IELC(NMSSVR)
     KY2 = IELC(KYOP2)
C
C    DEFINE KELOUT SWITCHES FOR STIFFNESS MATRIX
C
     KELOUT(1)=1
C
C  DEFINE KELREQ SWITCHES FOR LOAD VECTOR
```

```
C
   KELREQ(5) =1
C
C   --- GET THE SVR INDEX VECTOR
C
CALL SVGIDX (LOCSVR,SVINDX)
C
C   --- GET THE ELEMENT REAL CONSTANT DATA
C
C   REAL CONSTANT DATA IS INPUTTED USING SAME FORMAT
C   AS SOLID46 ELEMENT WITH KEYOPT(2) = 0
C   NL,LSYM,LP1,LP2,(BLANKS(2)),KREF,(BLANKS(5)),
C   MAT,THETA,TK FOR LAYER 1
C   MAT,THETA,TK FOR LAYER 2
C   ETC. UP TO LAYER NL
C
C   FOR THIS USER ELEMENT HOWEVER, THE VARIABLES
C   LSYM,LP1,LP2,BLANKS(2),KREF,BLANKS(5)
C   WILL NOT BE USED, AND WILL INPUTTED AS BLANKS
C
C
   CALL RVRGET (ELEM,IREAL,IELC(1),NRVR,RVR)C
C
   IF (KY2 .EQ. 0) THEN
    NUMLAYERS = RVR(1)
    K = 13
    KP1 = 14
    KP2 = 15
    DO 10 I = 1, NUMLAYERS
     MATNUM(I) = RVR(K)
     THETA = RVR(KP1)
     TK = RVR(KP2)
C
C   CONVERT ANSYS VARIABLES TO CHRISTENSEN VARIABLES
C
     ANGFIB(I,MATNUM(I)) = THETA
     THK(I) = TK
     K = K + 3
     KP1 = KP1 + 3
     KP2 = KP2 + 3
10   CONTINUE
    ELSE
     WRITE(IOTT,1)
1    FORMAT(/10X,'***** TAPERED LAYER OR MATRIX INPUT NOT',
   X /17X,'YET IMPLEMENTED ****** ',/)
     RETURN
    END IF
C
C   --- SET UP MATERIAL PROPERTIES
C
C   E11
C
   CALL PROPE1(ELEM,MATNUM(ELEM),1,0.0,PROPO)
   PROPS(1) = PROPO
C
```

```
C  E22
C
   CALL PROPE1(ELEM,MATNUM(ELEM),2,0.0,PROPO)
   PROPS(2) = PROPO
C
C  NU12
C
   CALL PROPE1(ELEM,MATNUM(ELEM),4,0.0,PROPO)
   PROPS(3) = PROPO
C
C  G12
C
   CALL PROPE1(ELEM,MATNUM(ELEM),7,0.0,PROPO)
   PROPS(4) = PROPO
C
C  G13
C
   CALL PROPE1(ELEM,MATNUM(ELEM),9,0.0,PROPO)
   PROPS(5) = PROPO
C
C  DENSITY
C
   CALL PROPE1(ELEM,MATNUM(ELEM),13,0.0,PROPO)
   PROPS(6) = PROPO
C
C  CHRIST.FOR VARIABLE P7 (LAMINA THICKNESS) IS NOW
C  INPUTTED AS A REAL CONSTANT IN ANSYS
C  THEREFORE PROPS(7) WILL BE SKIPPED
C  AND SET TO ZERO
C
   PROPS(7) = 0.0D0
C
C  ALPHAX
C
   CALL PROPE1(ELEM,MATNUM(ELEM),10,0.0,PROPO)
   PROPS(8) = PROPO
C
C  ALPHAY
C
   CALL PROPE1(ELEM,MATNUM(ELEM),11,0.0,PROPO)
   PROPS(9) = PROPO
C
C  ALPHAZ
C
   CALL PROPE1(ELEM,MATNUM(ELEM),12,0.0,PROPO)
   PROPS(10) = PROPO
C
C  CALCULATE THE STIFFNESS MATRIX
C
   CALL BRICK (ELEM,ZS,XYZ,NUMLAYERS,ANGFIB,PROPS,MATNUM)
C
C  WRITE OUT RESULTS
C
CALL STRBRI(U,ELEM,XYZ,PROPS,NR,NODES,NUMLAYERS,MATNUM,ANGFIB,
   X EDINDX,LCERST)
```

```
C
CALL TRACK (15, 'UEL102')
   RETURN
   END
C
C  ******************************************************************
C
   SUBROUTINE BRICK(IELEM,ESTIF,COORD,NLAYERS,IALPHA,PROPS,MATNO)
C
C  CALCULATES THE STIFFNESS MATRIX FROM THE INPUT DATA
C
DOUBLE PRECISION ESTIF(24,24),BMATX(6,42),DMATX(6,6)
   DOUBLE PRECISION SHAPE(14),DERIV(3,14),DBMAT(6,42),CARTD(3,14)
   DOUBLE PRECISION ELCOD(6,8),COORD(6,8),TSTIF(33,33) ,TEMP
   DOUBLE PRECISION POSGP(3), WEIGP(3), DVOLU
   COMMON /CONTROL/ MPOIN,NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,
   !  NPROP,NMATS,NVFIX,NEVAB,NINTR
   COMMON /FIXEDBC/ PRESC(100,3),NOFIX(100),IFPRE(100,3)
   DOUBLE PRECISION PROPS(10)
   COMMON /MESH/ LNODS(6,8),
   !    ISNODE(750),ISELEM(500)
   EXTERNAL TRACK,WRINQR
   INTEGER WRINQR,NLAYERS,IALPHA(100,100),MATNO(250),IELEM
   INTEGER NGAUS
   CALL TRACK (6,'BRICK')
   IOTT = WRINQR(2)
C
C
C  EVALUATE THE COORDINATES OF THE ELEMENT NODAL POINTS
C
C  --- SETS UP PARAMETERS FOR 8-NODED BRICK
C
   NNODE = 8
   NDIME = 3
C
C  --- SETS UP CHRIST.FOR ONLY VARIABLES
C
   NINTR = 1
   NEVAB = 24
   NGAUS = 3
   NSTRE = 6
   NDOFN = 3
   NPROP = 13
C
C  --- ZERO OUT VARIABLES
C
   EXISP = 0.0D0
   ETASP = 0.0D0
   RHOSP = 0.0D0
   DVOLU = 0.0D0
C
C  --- SET UP GAUSSIAN INTEGRATION CONSTRANTS
C
   CALL GAUSSQ(NGAUS,POSGP,WEIGP)
C
```

```
    DO 10 INODE = 1,NNODE
     DO 10 IDIME = 1,NDIME
    ELCOD(IDIME,INODE) = COORD(IDIME,INODE)
10   CONTINUE
C
C    INITIALIZE THE ELEMENT STIFFNESS MATRIX
C
    DO 20 IEVAB = 1,NEVAB+NINTR*9
    DO 20 JEVAB = 1,NEVAB+NINTR*9
    TSTIF(IEVAB,JEVAB) = 0.0D0
 20  CONTINUE
    KGASP = 0
C
C    ENTER LOOPS FOR AREA NUMERICAL INTEGRATION, STARTING WITH THE
C      LOOP FOR EACH LAYER IN THE ELEMENT
C
    DO 50 ILAYER = 1,NLAYERS
C
C    EVALUATE THE D-MATRIX IN THIS LAYER
C
    CALL DCHRIST(IELEM,ILAYER,IALPHA,PROPS,MATNO,DMATX)
C
C      NOW THE GAUSSIAN INTEGRATION OF X AND Y IN THIS LAYER
C
    DO 50 IGAUS = 1,NGAUS
    DO 50 JGAUS = 1,NGAUS
    KGASP = KGASP+1
    EXISP = POSGP(IGAUS)
    ETASP = POSGP(JGAUS)
    RHOSP = -1.+(2.*FLOAT(ILAYER)-1.)/FLOAT(NLAYERS)
C
C    EVALUATE THE SHAPE FUNCTIONS, ELEMENTAL VOLUME, ETC. AT THIS
C      SAMPLING POINT
C
    CALL SFBRIK(EXISP,ETASP,RHOSP,SHAPE,DERIV)
    CALL JACBRIK(IELEM,DJACB,CARTD,DERIV,ELCOD)
    DVOLU = DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)
   ! /FLOAT(NLAYERS)
C    EVALUATE THE B AND BXD MATRICES
C
    CALL BBRICK(BMATX,CARTD)
CALL DBE(BMATX,DMATX,DBMAT)
C
C    CALCULATE THE ELEMENT STIFFNESSES
C
DO 30 IEVAB = 1,NEVAB+NINTR*9
    DO 30 JEVAB = IEVAB,NEVAB+NINTR*9
    DO 30 ISTRE = 1,NSTRE
TSTIF(IEVAB,JEVAB) = TEMP +
   !    BMATX(ISTRE,IEVAB)*DBMAT(ISTRE,JEVAB)*DVOLU
30  CONTINUE
 50  CONTINUE
C
C    CONSTRUCT THE LOWER TRIANGLE OF THE STIFFNESS MATRIX
C
```

```
      DO 60 IEVAB = 1,NEVAB+NINTR*9
      DO 60 JEVAB = 1,NEVAB+NINTR*9
      TSTIF(JEVAB,IEVAB) = TSTIF(IEVAB,JEVAB)
   60 CONTINUE
C
C     NOW DO THE STATIC REDUCTION OF TSTIF TO ESTIF
C
      CALL REDUCE (TSTIF,ESTIF,NINTR)
C
C     ALL DONE
C
      CALL TRACK(16,'BRICK')
      RETURN
      END
C
C     ************************************************************
C
      SUBROUTINE REDUCE (TSTIF,ESTIF,NINTR)
C
C     PERFORMS A STATIC REDUCTION OF THE TWO INTERNAL DEGREES OF
C     FREEDOM IN THE 8-NODE BRICK THAT HAVE BEEN ADDED TO CAPTURE
C     THE THROUGH-THICKNESS DISPLACEMENT FIELD IN A LAMINATED
C     COMPOSITE MATERIAL
C
      EXTERNAL WRINQR
      INTEGER WRINQR
      DOUBLE PRECISION TSTIF(33,33),ESTIF(24,24),RSTIF(24,24),INDX(22)
      DOUBLE PRECISION STAU(22,24),STUA(24,22),STAA(22,22)
      DOUBLE PRECISION SINVAA(22,22),SINT(22,24)
      IOTT = WRINQR(2)
C
C     FIRST, GET THE PART OF THE MATRIX THAT HAS JUST THE
C     ADDED INTERNAL DEGREES OF FREEDOM
C
      NADD = 9*NINTR
C
C     --- INITALIZE ESTIF
C
      DO 90 I=1,24
      DO 90 J=1,24
   90 ESTIF(I,J) = 0.0D0
C
C     --- INITALIZE STAA
C
      DO 5 I = 1,NADD
      DO 5 J = 1,NADD
      STAA(I,J) = 0.0D0
    5 CONTINUE
      IF (NADD.GT.0) THEN
      DO 10 IAA = 1,NADD
      DO 10 KAA = 1,NADD
      STAA(IAA,KAA) = TSTIF(IAA+24,KAA+24)
   10 CONTINUE
C
C     NOW GET THE INVERSE OF THIS MATRIX FOR THE STATIC REDUCTION
```

```
C     EQUATION - NUMERICAL RECIPES IN FORTRAN - P. 38
C
      DO 20 IA = 1,NADD
      DO 15 JA = 1,NADD
          SINVAA(IA,JA) = 0.0D0
 15   CONTINUE
      SINVAA(IA,IA) = 1.0D0
 20   CONTINUE
      CALL LUDCMP(STAA,NADD,22,INDX,D)
      DO 25 JA = 1,NADD
          CALL LUBKSB(STAA,NADD,22,INDX,SINVAA(1,JA))
 25   CONTINUE
C
C     NOW DO THE MATRIX MULTIPLICATION OF [KUA] X [KAA INVERSE] X
C
C     FIRST GET [KUA] AND [KAU]
C
      DO 30 IROW = 1,NADD
      DO 30 ICOL = 1,24
          STAU(IROW,ICOL) = TSTIF(IROW+24,ICOL)
 30   CONTINUE
      DO 40 IROW = 1,24
      DO 40 ICOL = 1,NADD
          STUA(IROW,ICOL) = TSTIF(IROW,ICOL+24)
 40   CONTINUE
C
C     NOW MULTIPLY OUT THE MATRICES
C
      DO 50 IROW = 1,NADD
      DO 50 ICOL = 1 24
      SINT(IROW,ICOL) = 0.0D0
      DO 50 JROW = 1,NADD
      SINT(IROW,ICOL) = SINT(IROW,ICOL)
     ! +SINVAA(IROW,JROW)*STAU(JROW,ICOL)
 50   CONTINUE
      DO 60 IROW = 1,24
      DO 60 ICOL = 1,24
      RSTIF(IROW,ICOL) = 0.0D0
      DO 60 J = 1,NADD
      RSTIF(IROW,ICOL) = RSTIF(IROW,ICOL)
     ! +STUA(IROW,J)*SINT(J,ICOL)
 60   CONTINUE
C
C     NOW SUBTRACT RSTIF FROM TSTIF TO GET ESTIF
C
      DO 70 IROW = 1,24
      DO 70 ICOL = 1,24
      ESTIF(IROW,ICOL) = TSTIF(IROW,ICOL)-RSTIF(IROW,ICOL)
 70   CONTINUE
      ELSE
      DO 80 IROW = 1,24
      DO 80 ICOL = 1,24
      ESTIF(IROW,ICOL) = TSTIF(IROW,ICOL)
 80   CONTINUE
      ENDIF
```

```
C
C     ALL DONE
C
      RETURN
      END
C
C     ************************************************  ***********
C
      SUBROUTINE LUDCMP(A,N,NP,INDX,D)
C
C     ROUTINE EXTRACTED FROM "NUMERICAL RECIPES IN FORTRAN", PP. 35-36
C
      PARAMETER (NMAX=100,TINY=1.D-20)
      DOUBLE PRECISION A(NP,NP),INDX(NP),VV(NMAX),AAMAX
      EXTERNAL WRINQR
      INTEGER WRINQR,NP
      IOTT = WRINQR(2)
      D = 1.0D0
      DO 12 I=1,N
         AAMAX = 0.0D0
         DO 11 J = 1,N
      IF (ABS(A(I,J)).GT.AAMAX) AAMAX = ABS(A(I,J))
 11      CONTINUE
      IF (AAMAX.EQ.0.0D0) THEN
            WRITE (*,*) ' YOU HAVE A SINGULAR MATRIX'
            WRITE (8,*) A
            STOP
         ENDIF
         VV(I) = 1./AAMAX
 12   CONTINUE
      DO 19 J = 1,N
         DO 14 I = 1,J-1
            SUM = A(I,J)
            DO 13 K = 1,I-1
               SUM = SUM - A(I,K)*A(K,J)
 13         CONTINUE
            A(I,J) = SUM
 14      CONTINUE
         AAMAX = 0.0D0
         DO 16 I=J,N
            SUM = A(I,J)
            DO 15 K = 1,J-1
               SUM = SUM - A(I,K)*A(K,J)
 15         CONTINUE
            A(I,J) = SUM
            DUM = VV(I)*ABS(SUM)
            IF (DUM.GE.AAMAX) THEN
               IMAX = I
               AAMAX = DUM
            ENDIF
 16      CONTINUE
         IF (J.NE.IMAX) THEN
            DO 17 K = 1,N
               DUM = A(IMAX,K)
               A(IMAX,K) = A(J,K)
```

```
                  A(J,K) = DUM
17             CONTINUE
               D = -D
               VV(IMAX) = VV(J)
            ENDIF
            INDX(J) = IMAX
            IF (A(J,J).EQ.0.0D0) A(J,J) = TINY
            IF (J.NE.N) THEN
               DUM = 1./A(J,J)
               DO 18 I=J+1,N
                  A(I,J) = A(I,J)*DUM
18             CONTINUE
            ENDIF
19    CONTINUE
      RETURN
      END
C
C     ************************************************************
C
      SUBROUTINE LUBKSB(A,N,NP,INDX,B)
C
C     EXTRACTED FROM "NUMERICAL RECIPES IN FORTRAN"
C     DOES THE BACK SUBSTITUTION AFTER THE LU DECOMPOSITION
C
DOUBLE PRECISION A(NP,NP),INDX(NP),B(NP)
      INTEGER NP
C
      II = 0
      DO 12 I = 1,N
         LL = INDX(I)
         SUM = B(LL)
         B(LL) = B(I)
         IF (II.NE.0) THEN
            DO 11 J = II,I-1
               SUM = SUM - A(I,J)*B(J)
11          CONTINUE
         ELSEIF (SUM.NE.0.0D0) THEN
               II = I
         ENDIF
         B(I) = SUM
12    CONTINUE
      DO 14 I = N,1,-1
         SUM = B(I)
         IF (I.LT.N) THEN
            DO 13 J = I+1,N
               SUM = SUM - A(I,J)*B(J)
13          CONTINUE
         ENDIF
         B(I) = SUM/A(I,I)
14    CONTINUE
      RETURN
      END
C
C     ************************************************************
C
```

```
      SUBROUTINE DBE(BMATX,DMATX,DBMAT)
C
C     CALCULATES D X B
C
      DOUBLE PRECISION BMATX(6,42),DMATX(6,6),DBMAT(6,42)
      COMMON /CONTROL/ MPOIN,NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,
     !     NPROP,NMATS,NVFIX,NEVAB,NINTR
C
      DO 10 ISTRE = 1,NSTRE
      DO 10 IEVAB = 1,NEVAB+9*NINTR
      DBMAT(ISTRE,IEVAB) = 0.0D0
      DO 10 JSTRE = 1,NSTRE
      DBMAT(ISTRE,IEVAB) = DBMAT(ISTRE,IEVAB)+
     !   DMATX(ISTRE,JSTRE)*EMATX(JSTRE,IEVAB)
   10 CONTINUE
      RETURN
      END
C
C     ************************************************************
C
      SUBROUTINE SFBRIK(S,T,R,SHAPE,DERIV)
C
C     CALCULATES SHAPE FUNCTIONS AND THEIR DERIVATIVES FOR
C     BRICK STRESS AND STRAIN 2-D ELEMENTS
C
      DOUBLE PRECISION SHAPE(14),DERIV(3,14)
      EXTERNAL WRINQR
      INTEGER WRINQR
C
C     --- INITIALIZE SHAPE FUNCTIONS AND DERIVATIVES
C
      IOTT=WRINQR(2)
      DO 10 I = 1,14
   10 SHAPE(I) = 0.0D0
      DO 20 I = 1,3
      DO 20 J = 1,14
   20 DERIV(I,J)=0.0D0
C
C
      SHAPE(1) = 0.125*(1.-T)*(1.-S)*(1.-R)
      SHAPE(2) = 0.125*(1.-T)*(1.+S)*(1.-R)
      SHAPE(3) = 0.125*(1.+T)*(1.+S)*(1.-R)
      SHAPE(4) = 0.125*(1.+T)*(1.-S)*(1.-R)
      SHAPE(5) = 0.125*(1.-T)*(1.-S)*(1.+R)
      SHAPE(6) = 0.125*(1.-T)*(1.+S)*(1.+R)
      SHAPE(7) = 0.125*(1.+T)*(1.+S)*(1.+R)
      SHAPE(8) = 0.125*(1.+T)*(1.-S)*(1.+R)
      SHAPE(9) = 1.-S**2
      SHAPE(10) = 1.-T**2
      SHAPE(11) = 1.-R**2
      SHAPE(12) = S-S**3
      SHAPE(13) = T-T**3
      SHAPE(14) = R-R**3
C
```

```
C   AND THEIR DERIVATIVES
C
      DERIV(1,1) = -.125*(1.-T)*(1.-R)
      DERIV(1,2) = .125*(1.-T)*(1.-R)
      DERIV(1,3) = .125*(1.+T)*(1.-R)
      DERIV(1,4) = -.125*(1.+T)*(1.-R)
      DERIV(1,5) = -.125*(1.-T)*(1.+R)
      DERIV(1,6) = .125*(1.-T)*(1.+R)
      DERIV(1,7) = .125*(1.+T)*(1.+R)
      DERIV(1,8) = -.125*(1.+T)*(1.+R)
        DERIV(1,9) = -2.*S
        DERIV(1,10) = 0.0D0
        DERIV(1,11) = 0.0D0
        DERIV(1,12) = 1.-3.*S**2
        DERIV(1,13) = 0.0D0
        DERIV(1,14) = 0.0D0
      DERIV(2,1) = -.125*(1.-S)*(1.-R)
      DERIV(2,2) = -.125*(1.+S)*(1.-R)
      DERIV(2,3) = .125*(1.+S)*(1.-R)
      DERIV(2,4) = .125*(1.-S)*(1.-R)
      DERIV(2,5) = -.125*(1.-S)*(1.+R)
      DERIV(2,6) = -.125*(1.+S)*(1.+R)
      DERIV(2,7) = .125*(1.+S)*(1.+R)
      DERIV(2,8) = .125*(1.-S)*(1.+R)
        DERIV(2,9) = 0.0D0
        DERIV(2,10) = -2.*T
        DERIV(2,11) = 0.0D0
        DERIV(2,12) = 0.0D0
        DERIV(2,13) = 1.-3.*T**2
        DERIV(2,14) = 0.0D0
      DERIV(3,1) = -.125*(1.-S)*(1.-T)
      DERIV(3,2) = -.125*(1.+S)*(1.-T)
      DERIV(3,3) = -.125*(1.+S)*(1.+T)
      DERIV(3,4) = -.125*(1.-S)*(1.+T)
      DERIV(3,5) = .125*(1.-S)*(1.-T)
      DERIV(3,6) = .125*(1.+S)*(1.-T)
      DERIV(3,7) = .125*(1.+S)*(1.+T)
      DERIV(3,8) = .125*(1.-S)*(1.+T)
        DERIV(3,9) = 0.0D0
        DERIV(3,10) = 0.0D0
        DERIV(3,11) = -2.*R
        DERIV(3,12) = 0.0D0
        DERIV(3,13) = 0.0D0
        DERIV(3,14) = 1.-3.*R**2
C
C   ALL DONE
C
      RETURN
      END
C
C   ************************************************************
C
      SUBROUTINE JACBRIK(IELEM,DJACB,CARTD,DERIV,ELCOD)
C
C   CALCULATES THE JACOBIAN MATRIX AND DETERMINANT AND INVERSE
```

```
C     FOR 3-D ELEMENTS
C
DOUBLE PRECISION XM(3,3),XI(3,3),CARTD(3,14)

C
   COMMON /CONTROL/ MPOIN,NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,
   !     NPROP,NMATS,NVFIX,NEVAB,NINTR

   DOUBLE PRECISION DERIV(3,14),ELCOD(6,8)
   INTEGER IELEM
   EXTERNAL WRINQR
   INTEGER WRINQR
   IOTT = WRINQR(2)
C
C   --- INITIALIZE XI MATRIX
C
   DO 10 I = 1, NDIME
   DO 10 J = 1, NDIME
   XM(I,J) = 0.0D0
10  XI(I,J) = 0.0D0
   DJACB = 0.0D0
C
C   CREATE JACOBIAN MATRIX
C
   DO 20 IDIME = 1,NDIME
   DO 20 JDIME = 1,NDIME
DO 20 INODE = 1,NNODE
   XM(IDIME,JDIME) = XM(IDIME,JDIME)+
   !    DERIV(IDIME,INODE)*ELCOD(JDIME,INODE)
 20 CONTINUE
C
C   NOW THE DETERMINANT AND INVERSE OF THE JACOBIAN
C
   DJACB = XM(1,1)*(XM(2,2)*XM(3,3)-XM(3,2)*XM(2,3))
   !    -XM(1,2)*(XM(2,1)*XM(3,3)-XM(3,1)*XM(2,3))
   !    +XM(1,3)*(XM(2,1)*XM(3,2)-XM(3,1)*XM(2,2))
   IF (DJACB.GT.0.) GO TO 30
   WRITE (IOTT,999) IELEM
 999 FORMAT (//,' ELEMENT NUMBER ',I5,' HAS ZERO OR ',
   !      'NEGATIVE AREA - PLEASE CHECK INPUT',////)
   STOP
 30 XI(1,1) = (XM(2,2)*XM(3,3)-XM(3,2)*XM(2,3))/DJACB
   XI(1,2) = (XM(3,2)*XM(1,3)-XM(1,2)*XM(3,3))/DJACB
   XI(1,3) = (XM(1,2)*XM(2,3)-XM(2,2)*XM(1,3))/DJACB
   XI(2,1) = (XM(3,1)*XM(2,3)-XM(2,1)*XM(3,3))/DJACB
   XI(2,2) = (XM(1,1)*XM(3,3)-XM(3,1)*XM(1,3))/DJACB
   XI(2,3) = (XM(2,1)*XM(1,3)-XM(1,1)*XM(2,3))/DJACB
   XI(3,1) = (XM(2,1)*XM(3,2)-XM(3,1)*XM(2,2))/DJACB
   XI(3,2) = (XM(3,1)*XM(1,2)-XM(1,1)*XM(3,2))/DJACB
   XI(3,3) = (XM(1,1)*XM(2,2)-XM(2,1)*XM(1,2))/DJACB
C
C   CALCULATE CARTESIAN DERIVATIVES
C
   DO 50 I=1,6
   DO 50 J=1,42
```

```
50   CARTD(I,J) = 0.0D0
     DO 40 IDIME = 1,NDIME
     DO 40 INODE = 1,NNODE+NINTR*3
C      CARTD(IDIME,INODE) = 0.0D0
     DO 40 JDIME = 1,NDIME
     CARTD(IDIME,INODE) = CARTD(IDIME,INODE)+
    !   XI(IDIME,JDIME)*DERIV(JDIME,INODE)
 40  CONTINUE
C
C    ALL DONE
C
     RETURN
     END
C
C    **********************************************************
C
     SUBROUTINE BBRICK(BMATX,CARTD)
C
C    CALCULATES THE STRAIN MATRIX B FOR THE BRICK ELEMENT
C
     DOUBLE PRECISION BMATX(6,42),CARTD(3,14)
     COMMON /CONTROL/ MPOIN,NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,
    !     NPROP,NMATS,NVFIX,NEVAB,NINTR
C
     DO 5 I=1,6
     DO 5 J = 1,42
5    BMATX(I,J) = 0.0D0
     DO 10 INODE = 1,NNODE+NINTR*3
     MGASH = (INODE-1)*3+1
     NGASH = (INODE-1)*3+2
     LGASH = (INODE-1)*3+3
     BMATX(1,MGASH) = CARTD(1,INODE)
     BMATX(1,NGASH) = 0.0D0
     BMATX(1,LGASH) = 0.0D0
     BMATX(2,MGASH) = 0.0D0
     BMATX(2,NGASH) = CARTD(2,INODE)
     BMATX(2,LGASH) = 0.0D0
     BMATX(3,MGASH) = 0.0D0
     BMATX(3,NGASH) = 0.0D0
     BMATX(3,LGASH) = CARTD(3,INODE)
     BMATX(4,MGASH) = CARTD(2,INODE)
     BMATX(4,NGASH) = CARTD(1,INODE)
     BMATX(4,LGASH) = 0.0D0
     BMATX(5,MGASH) = 0.0D0
     BMATX(5,NGASH) = CARTD(3,INODE)
     BMATX(5,LGASH) = CARTD(2,INODE)
     BMATX(6,MGASH) = CARTD(3,INODE)
     BMATX(6,NGASH) = 0.0D0
     BMATX(6,LGASH) = CARTD(1,INODE)
 10  CONTINUE
C
C    ALL DONE
C
     RETURN
     END
```

```
C
C   ******************************************************
C
      SUBROUTINE DCHRIST(IELEM,ILAYER,IALPHA,PROPS,MATNO,DMATX)
C
C   ROUTINE TO EVALUATE ELASTICITY MATRIX FOR THE ITH LAYER OF
C     THE ELEMENT, ACCORDING TO CHRISTENSEN (1990) EQUATION 25
C
      COMMON /CONTROL/ MPOIN,NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,
     !     NPROP,NMATS,NVFIX,NEVAB,NINTR
      COMMON /FIXEDBC/ PRESC(100,3),NOFIX(100),IFPRE(100,3)
      INTEGER  IALPHA(100,100),MATNO(250),IELEM
      DOUBLE PRECISION PROPS(10),AVEMU,AVELAM,AVEMOD,A1,A2
      COMMON /MESH/ LNODS(6,8),
     !     ISNODE(750),ISELEM(500)
C
      DOUBLE PRECISION DMATX(6,6)
C
C
C   MATERIAL PROPERTIES INPUT AS FOLLOWS
C
C   P1 = E11
C   P2 = E22
C   P3 = NU12
C   P4 = G12
C   P5 = G13
C   P6 = DENSITY
C     P7 = LAMINA THICKNESS
C     P8 = THERMAL EXPANSION - X
C     P9 = THERMAL EXPANSION - Y
C     P10 = THERMAL EXPANSION - Z
C
C     CALCULATE THE MATRIX DOMINATED PROPERTIES AND THE AXIAL PROPERTY
C
C
C --- ZERO OUT VARIABLES
C
      AVEMU = 0.0D0
      AVELAM = 0.0D0
      AVEMOD = 0.0D0
      A1 = 0.0D0
      A2 = 0.0D0
C
       IF (PROPS(5).LE.1.) THEN
          AVEMU = (PROPS(2)*(1.-PROPS(3))/(2.*(1.-PROPS(2)/
     !        PROPS(1)*PROPS(3)**2))+PROPS(4))/2.
      ELSE
          AVEMU = (PROPS(2)*(1.-PROPS(3))/(1.-PROPS(2)/PROPS(1)*
     !        PROPS(3)**2)+2.*PROPS(4)+PROPS(5))/5.
      ENDIF
      AVELAM = AVEMU*2.*PROPS(3)/(1.-2.*PROPS(3))
       AVEMOD = PROPS(1)-AVEMU*2.*(1.+PROPS(3))
C
C   ZERO OUT THE D-MATRIX
C
```

```
      DO 10 ISTRE = 1,NSTRE
      DO 10 JSTRE = 1,NSTRE
      DMATX(ISTRE,JSTRE) = 0.0D0
   10 CONTINUE
C
C     EVALUATE THE DIRECTION COSINES FOR THIS LAYER
C
       CONVER = 3.1415926536/180.
       A1 = COS(CONVER*FLOAT(IALPHA(ILAYER,MATNO(IELEM))))
       A2 = SIN(CONVER*FLOAT(IALPHA(ILAYER,MATNO(IELEM))))
C     NOW THE COMPONENTS OF THE D-MATRIX FOR THIS LAYER
C
       DMATX(1,1) = AVELAM*(1.-PROPS(3))/PROPS(3) + AVEMOD*A1**4
       DMATX(1,2) = AVELAM + AVEMOD*A1**2*A2**2
       DMATX(1,3) = AVELAM
       DMATX(1,4) = AVEMOD*A1**3*A2
       DMATX(2,1) = DMATX(1,2)
       DMATX(2,2) = AVELAM*(1.-PROPS(3))/PROPS(3) + AVEMOD*A2**4
       DMATX(2,3) = AVELAM
       DMATX(2,4) = AVEMOD*A1*A2**3
       DMATX(3,1) = DMATX(1,3)
       DMATX(3,2) = DMATX(2,3)
       DMATX(3,3) = AVELAM*(1.-PROPS(3))/PROPS(3)
       DMATX(4,1) = DMATX(1,4)
       DMATX(4,2) = DMATX(2,4)
       DMATX(4,4) = AVEMU + AVEMOD*A1**2*A2**2
       DMATX(5,5) = AVEMU
       DMATX(6,6) = AVEMU
C
C    ALL DONE
C
      RETURN
      END
C
C    ***********************************************************
C
      SUBROUTINE GAUSSQ(NGAUS,POSGP,WEIGP)
C
C    ROUTINE TO SET UP SAMPLING POINTS AND WEIGHTING FACTORS
C    FOR THE ELEMENT NUMERICAL INTEGRATIONS
C
      COMMON /CONTROL/ MPOIN,NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,
     !    NPROP,NMATS,NVFIX,NEVAB,NINTR
      DOUBLE PRECISION POSGP(3),WEIGP(3)
      INTEGER KGAUS,IGAUS,JGAUS
C
      EXTERNAL TRACK,WRINQR
      INTEGER WRINQR,NGAUS
      IOTT = WRINQR(2)
C
      CALL TRACK(7,'GAUSSQ')
      IF (NGAUS.GT.2) GO TO 10
      POSGP(1) = -.577350269189626
      WEIGP(1) = 1.0D0
      GO TO 20
```

```
 10 POSGP(1) = .774596669241483
C
C   CHANGE BY M. JONES ON 1/15/96
C   CHANGED POSGP(3) TO POSGP(2) AND WEIGP(3) TO WEIGP(2)
C   TO REFLECT GUASS-QUAD SAMPLING POINTS AND WEIGHTS
C   WHEN KGUAS = 3
C
   POSGP(2) = 0.0D0
   WEIGP(1) = .555555555555556
   WEIGP(2) = 0.0D0
 20 KGAUS = NGAUS/2
   DO 30 IGASH = 1,KGAUS
   JGASH = NGAUS+1-IGASH
   POSGP(JGASH) = -POSGP(IGASH)
   WEIGP(JGASH) = WEIGP(IGASH)
 30 CONTINUE
   CALL TRACK(17,'GAUSSQ')
   RETURN
   END
C
C
C
C   ***********************************************************
C
   SUBROUTINE STRBRI(ASDIS,IELEM,COORD,PROPS,NR,LNODS,NLAYERS,
  X MATNO,IALPHA,LCERST,EDINDX)
C
C   ROUTINE TO CALCULATE STRESSES IN BRICK ELEMENTS AFTER
C   DISPLACEMENTS HAVE BEEN CALCULATED
C
C    LOGICAL THERMAL
C         DOUBLE PRECISION STRIN(6,8),STRAN(6,8)
   EXTERNAL WRINQR,UEP102,TRACK,ELDWRT
   INTEGER WRINQR,NLAYERS,IALPHA(100,100),MATNO(250),IELEM,NR
   INTEGER NGAUS,SVINDX(1)
   DOUBLE PRECISION ELDIS(3,8),STRES(6),STEMP(27,6),STRAI(6),
  X SRTMP(27,6),SMATX(6,24,27),SRMAT(6,24,27),BMATX(6,42),
  X DMATX(6,6)
   DOUBLE PRECISION DBMAT(6,42),CARTD(3,14),COORD(6,8),ELCOD(6,8)
   DOUBLE PRECISION SHAPE(14),DERIV(3,14),PROPS(10)
   DOUBLE PRECISION ASDIS(NR,5)
   DOUBLE PRECISION POSGP(3),WEIGP(3)
   INTEGER EDINDX(20),LCERST
   INTEGER LNODS(8)
   INCLUDE 'ELPARM'
C
   COMMON /CONTROL/ MPOIN,NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,
  !     NPROP,NMATS,NVFIX,NEVAB,NINTR
   COMMON /FIXEDBC/ PRESC(100,3),NOFIX(100),IFPRE(100,3),
  !     ISNODE(750),ISELEM(500)
C    KGAST = 0
C    THERMAL = .FALSE.
C
NNODE = 8
   NDIME = 3
```

```
      NINTR = 1
      NEVAB = 24
      NGAUS = 3
      NSTRE = 6
      NDOFN = 3
      NPROP = 13
C     WRITE HEADER
C
      IOTT = WRINQR(2)
      WRITE (IOTT,*) ' CALCULATING STRESSES'
      WRITE (6,2)
 2    FORMAT (/,'   ELEM',2X,
     !  '   X      Y     Z',/,'   SX       SY       SZ',
     !  '       SXY       SYZ       SXZ')
C
C     CALCULATE AND OUTPUT STRESSES FOR EACH ELEMENT
C
         KGAST = 0
C
C     EVALUATE THE COORDINATES OF THE ELEMENT NODES
C
C
C
      DO 80 INODE = 1,NNODE
C       LNODE = IABS(LNODS(IELEM,INODE))
      DO 80 IDIME = 1,NDIME
      ELCOD(IDIME,INODE) = COORD(IDIME,INODE)
80    CONTINUE
      XCENT = 0.0D0
      YCENT = 0.0D0
C
C     --- Z-HEIGHT OF ELEMENT
C
          DZELEM = ELCOD(3,5)-ELCOD(3,1)
          Z1 = ELCOD(3,1)
      DO 9 INODE = 1,NNODE
         XCENT = XCENT + ELCOD(1,INODE)*.125
         YCENT = YCENT + ELCOD(2,INODE)*.125
 9    CONTINUE
C
C     IDENTIFY THE DISPLACEMENTS OF THE ELEMENT NODAL POINTS
C
      NPOSN = 0
      DO 10 INODE = 1,NNODE
      DO 10 IDOFN = 1,NDOFN
      NPOSN = NPOSN+1
      ELDIS(IDOFN,INODE) = ASDIS(NPOSN,1)
 10   CONTINUE
C
C     CALCULATE THE STRESS AND STRAIN MATRIX FOR THE ELEMENT
C
          DO 90 ILAYER = 1,NLAYERS
      RHOSP = -1.+(2.*FLOAT(ILAYER)-1.)/FLOAT(NLAYERS)
          ZCENT = Z1 + .5*(1.+RHOSP)*DZELEM
C
```

```
C   ZERO OUT THE TEMPORARY STRESSES
C
    NGASP = NGAUS*NGAUS
    DO 5 JSTRE = 1,NSTRE
    DO 5 IGASP = 1,NGASP
    STEMP(IGASP,JSTRE) = 0.
    SRTMP(IGASP,JSTRE) = 0.
  5 CONTINUE
C
C   EVALUATE THE D-MATRIX
C
        CALL DCHRIST(IELEM,ILAYER,IALPHA,PROPS,MATNO,DMATX)
C
        KGASP = 0
        DO 40 IGAUS = 1,NGAUS
    DO 40 JGAUS = 1,NGAUS
    KGASP = KGASP+1
    EXISP = POSGP(IGAUS)
    ETASP = POSGP(JGAUS)
C
C   EVALUATE THE SHAPE FUNCTIONS, ELEMENTAL VOLUME, ETC.
C
    CALL SFBRIK(EXISP,ETASP,RHOSP,SHAPE,DERIV)
    CALL JACBRIK(IELEM,DJACB,CARTD,DERIV,ELCOD)
C
C   EVALUATE THE B AND BXD MATRICES
C
    CALL BBRICK(BMATX,CARTD)
     CALL DBE(BMATX,DMATX,DBMAT)
C
C   STORE THE COMPONENTS OF THE STRESS AND STRAIN MARTICES
C
    DO 15 ISTRE = 1,NSTRE
    DO 15 IEVAB = 1,NEVAB
    SMATX(ISTRE,IEVAB,KGASP) = DBMAT(ISTRE,IEVAB)
    SRMAT(ISTRE,IEVAB,KGASP) = BMATX(ISTRE,IEVAB)
 15 CONTINUE
 40 CONTINUE
C
C   NOW TO CALCULATE STRESSES
C
    KGASP = 0
C
C   ENTER LOOPS OVER EACH GAUSS POINT IN THE LAYER
C
    DO 50 IGAUS = 1,NGAUS
    DO 50 JGAUS = 1,NGAUS
    KGAST = KGAST+1
    KGASP = KGASP+1
C
C   COMPUTE THE STRESS AND STRAIN COMPONENTS AT THE SAMPLING
C   POINTS
C
    DO 20 ISTRE = 1,NSTRE
    KGASH = 0
```

```
      DO 20 INODE = 1,NNODE
      DO 20 IDOFN = 1,NDOFN
      KGASH = KGASH+1
      STEMP(KGASP,ISTRE) = STEMP(KGASP,ISTRE)+
     !   SMATX(ISTRE,KGASH,KGASP)*ELDIS(IDOFN,INODE)
      SRTMP(KGASP,ISTRE) = SRTMP(KGASP,ISTRE)+
     !   SRMAT(ISTRE,KGASH,KGASP)*ELDIS(IDOFN,INODE)
   20 CONTINUE
C
C     FOR THERMAL LOADING, ADD ON THE INITIAL THERMAL STRESS
C
C      IF (THERMAL) THEN
C      READ (1,REC=IELEM) STRIN,STRAN
C      DO 30 ISTR1 = 1,NSTRE
C      STEMP(KGASP,ISTR1) = STEMP(KGASP,ISTR1)+STRIN(ISTR1,KGAST)
C      SRTMP(KGASP,ISTR1) = SRTMP(KGASP,ISTR1)+STRAN(ISTR1,KGAST)
C 30   CONTINUE
C      ENDIF
   50 CONTINUE
    3 FORMAT (2I4,3F10.4,2(/,1X,6G12.5))
C
C     OUTPUT STRESSES FOR THIS LAYER
C
C
C  --- AT THIS POINT UEP102 SHOULD BE CALLED FOR OUTPUT
C
      NGASP = NGAUS*NGAUS
           DGASP = 2.*FLOAT(NGASP)
      DO 60 LSTRE=1,NSTRE
      STRES(LSTRE) = 0.
      STRAI(LSTRE) = 0.
   60 CONTINUE
      DO 70 KSTRE = 1,NSTRE
      DO 70 IGASP = 1,NGASP
      STRES(KSTRE) = STRES(KSTRE)+STEMP(IGASP,KSTRE)/DGASP
      STRAI(KSTRE) = STRAI(KSTRE)+SRTMP(IGASP,KSTRE)/DGASP
   70 CONTINUE
   90 WRITE (IOTT,3) IELEM,ILAYER,XCENT,YCENT,ZCENT,(STRES(ISTRE),
     !     ISTRE=1,NSTRE),(STRAI(ISTRE),ISTRE=1,NSTRE)
   CALL ELDWRT(IELM,EDENS,LCERST,EDINDX(1),NSTRE,STRES)
   RETURN
    END
```

APPENDIX B

First Row of Elements (45,-45 layers)

```
     ELMDAT( 1) =  1 for Element  45
 CALCULATING STRESSES

  ELEM    X     Y   Z
   SX      SY      SZ      SXY      SYZ      SXZ
 45  1   0.5000  -0.0500   0.0250
  85665.    -487.81    481.11    24702.    82.856    8328.7
 0.30951E-01-0.20127E-01-0.26677E-02-0.44800E-02 0.98246E-04 0.98758E-02
 45  2   0.5000  -0.0500   0.0750
  87187.    1504.4    521.47    -26619.    -69.370    8306.2
 0.30844E-01-0.19955E-01-0.26677E-02 0.41761E-02-0.82256E-04 0.98491E-02

     ELMDAT( 1) =  1 for Element  39
 CALCULATING STRESSES

  ELEM    X     Y   Z
   SX      SY      SZ      SXY      SYZ      SXZ
 39  1   0.5000  -0.1500   0.0250
  91559.    687.80    -355.99    35803.    49.614    563.02
 0.30953E-01-0.22922E-01-0.22899E-02-0.48002E-04 0.58830E-04 0.66760E-03
 39  2   0.5000  -0.1500   0.0750
  89013.    -2093.8    -402.32    -32834.    -116.77    559.10
 0.30985E-01-0.23030E-01-0.22899E-02 0.54083E-03-0.13846E-03 0.66295E-03


     ELMDAT( 1) =  1 for Element  33
 CALCULATING STRESSES

  ELEM    X     Y   Z
   SX      SY      SZ      SXY      SYZ      SXZ
 33  1   0.5000  -0.2500   0.0250
  91228.    -503.58    43.513    34929.    -43.803    50.369
 0.31176E-01-0.23210E-01-0.20985E-02-0.15644E-03-0.51939E-04 0.59726E-04
 33  2   0.5000  -0.2500   0.0750
  92541.    511.04    46.429    -36300.    -38.825    45.634
 0.31266E-01-0.23296E-01-0.20985E-02-0.95467E-04-0.46037E-04 0.54111E-04
```

ELMDAT(1) = 1 for Element 27
CALCULATING STRESSES

ELEM X Y Z
SX SY SZ SXY SYZ SXZ
27 1 0.5000 -0.3500 0.0250
92021. 192.20 60.987 35598. 15.948 -0.80817
0.31249E-01-0.23195E-01-0.21144E-02-0.10541E-03 0.18911E-04-0.95829E-06
27 2 0.5000 -0.3500 0.0750
93056. 1234.2 69.668 -36795. 24.362 -10.832
0.31254E-01-0.23186E-01-0.21144E-02-0.10709E-03 0.28887E-04-0.12844E-04

ELMDAT(1) = 1 for Element 21
CALCULATING STRESSES

ELEM X Y Z
SX SY SZ SXY SYZ SXZ
21 1 0.5000 -0.4500 0.0250
92654. 999.15 69.529 36388. 18.344 -1.2282
0.31210E-01-0.23130E-01-0.21176E-02 0.20634E-04 0.21751E-04-0.14564E-05
21 2 0.5000 -0.4500 0.0750
92285. 732.56 48.222 -36100 12.680 -12.178
0.31162E-01-0.23117E-01-0.21176E-02 0.40114E-05 0.15035E-04-0.14440E-04

ELMDAT(1) = 1 for Element 15
CALCULATING STRESSES

ELEM X Y Z
SX SY SZ SXY SYZ SXZ
15 1 0.5000 -0.5500 0.0250
92654. 1148.9 82.563 36468. 27.817 -2.3970
0.31167E-01-0.23085E-01-0.21125E-02 0.33844E-04 0.32984E-04-0.28422E-05
15 2 0.5000 -0.5500 0.0750
91328. -39.729 -12.260 -35374. 29.173 -9.7503
0.31048E-01-0.23121E-01-0.21125E-02 0.40684E-04 0.34592E-04-0.11561E-04

ELMDAT(1) = 1 for Element 9
CALCULATING STRESSES

```
 ELEM    X     Y    Z
  SX       SY       SZ       SXY        SYZ         SXZ
 9  1   0.5000  -0.6500   0.0250
 91468.    411.51    -311.82    35112.    -172.14    -443.69
0.31105E-01-0.22880E-01-0.23220E-02-0.34010E-03-0.20412E-03-0.52610E-03
 9  2   0.5000  -0.6500   0.0750
 90467.    -266.98    -471.66    -34788.    3.1402    -438.84
0.30879E-01-0.22916E-01-0.23220E-02 0.18055E-03 0.37235E-05-0.52036E-03
```

ELMDAT(1) = 1 for Element 3
CALCULATING STRESSES

```
 ELEM    X     Y    Z
  SX       SY       SZ       SXY        SYZ         SXZ
 3  1   0.5000  -0.7500   0.0250
 86481.    279.66    250.74    25474.    -189.82    -8152.0
0.31016E-01-0.20091E-01-0.27950E-02-0.44209E-02-0.22508E-03-0.96663E-02
 3  2   0.5000  -0.7500   0.0750
 85291.    -459.24    51.317    -25167.    33.788    -8122.8
0.30719E-01-0.20120E-01-0.27950E-02 0.42034E-02 0.40064E-04-0.96317E-02
```

Second Row of Elements (-45,45 layers)

ELMDAT(2) = 2 for Element 93
CALCULATING STRESSES

```
 ELEM    X     Y    Z
  SX       SY       SZ       SXY        SYZ         SXZ
93  1   0.5000  -0.0500   0.1250
 87187.    1504.4    521.47    -26619.    69.370    -8306.2
0.30844E-01-0.19955E-01-0.26677E-02 0.41761E-02 0.82256E-04-0.98491E-02
93  2   0.5000  -0.0500   0.1750
 85665.    -487.81    481.11    24702.    -82.856    -8328.7
0.30951E-01-0.20127E-01-0.26677E-02-0.44800E-02-0.98246E-04-0.98758E-02
```

ELMDAT(2) = 2 for Element 87
CALCULATING STRESSES

ELEM X Y Z
SX SY SZ SXY SYZ SXZ
87 1 0.5000 -0.1500 0.1250
89013. -2093.8 -402.32 -32834. 116.77 -559.10
0.30985E-01-0.23030E-01-0.22899E-02 0.54083E-03 0.13846E-03-0.66295E-03
87 2 0.5000 -0.1500 0.1750
91559. 687.80 -355.99 35803. -49.614 -563.02
0.30953E-01-0.22922E-01-0.22899E-02-0.48002E-04-0.58830E-04-0.66760E-03

ELMDAT(2) = 2 for Element 81
CALCULATING STRESSES

ELEM X Y Z
SX SY SZ SXY SYZ SXZ
81 1 0.5000 -0.2500 0.1250
92541. 511.04 46.429 -36300. 38.825 -45.634
0.31266E-01-0.23296E-01-0.20985E-02-0.96467E-04 0.45037E-04-0.54111E-04
81 2 0.5000 -0.2500 0.1750
91228. -503.58 43.513 34929. 43.803 -50.369
0.31176E-01-0.23210E-01-0.20985E-02-0.15644E-03 0.51939E-04-0.59726E-04

ELMDAT(2) = 2 for Element 75
CALCULATING STRESSES

ELEM X Y Z
SX SY SZ SXY SYZ SXZ
75 1 0.5000 -0.3500 0.1250
93056. 1234.2 69.668 -36795. -24.362 10.832
0.31254E-01-0.23186E-01-0.21144E-02-0.10709E-03-0.28837E-04 0.12844E-04
75 2 0.5000 -0.3500 0.1750
92021. 192.20 60.987 35598. -15.948 0.80817
0.31249E-01-0.23195E-01-0.21144E-02-0.10541E-03-0.18911E-04 0.95829E-06

ELMDAT(2) = 2 for Element 69
CALCULATING STRESSES

```
 ELEM    X     Y   Z
  SX      SY     SZ     SXY      SYZ      SXZ
69  1   0.5000  -0.4500   0.1250
 92285.    732.56    48.222    -36100.    -12.680     12.178
0.31162E-01-0.23117E-01-0.21176E-02 0.40114E-05-0.15035E-04 0.14440E-04
69  2   0.5000  -0.4500   0.1750
 92654.    999.15    69.529    36388.    -18.344     1.2282
0.31210E-01-0.23130E-01-0.21176E-02 0.20634E-04-0.21751E-04 0.14564E-05
```

ELMDAT(2) = 2 for Element 63
CALCULATING STRESSES

```
 ELEM    X     Y   Z
  SX      SY     SZ     SXY      SYZ      SXZ
63  1   0.5000  -0.5500   0.1250
 91328.   -39.729   -12.260   -35374.    -29.173     9.7503
0.31048E-01-0.23121E-01-0.21125E-02 0.40684E-04-0.34592E-04 0.11561E-04
63  2   0.5000  -0.5500   0.1750
 92654.    1148.9    82.563    36468.    -27.817     2.3970
0.31167E-01-0.23085E-01-0.21125E-02 0.33844E-04-0.32984E-04 0.28422E-05
```

ELMDAT(2) = 2 for Element 57
CALCULATING STRESSES

```
 ELEM    X     Y   Z
  SX      SY     SZ     SXY      SYZ      SXZ
57  1   0.5000  -0.6500   0.1250
 90467.   -266.98   -471.66   -34788.    -3.1402     438.84
0.30879E-01-0.22916E-01-0.23220E-02 0.18055E-03-0.37235E-05 0.52036E-03
57  2   0.5000  -0.6500   0.1750
 91468.    411.51   -311.82    35112.     172.14     443.69
0.31105E-01-0.22880E-01-0.23220E-02-0.34010E-03 0.20412E-03 0.52610E-03
```

ELMDAT(2) = 2 for Element 51
CALCULATING STRESSES

```
 ELEM    X     Y   Z
  SX      SY      SZ      SXY      SYZ       SXZ
51  1   0.5000  -0.7500   0.1250
 85291.   -459.24    51.317   -25167.   -33.788    8122.8
0.30719E-01-0.20120E-01-0.27950E-02 0.42034E-02-0.40064E-04 0.96317E-02
51  2   0.5000  -0.7500   0.1750
 86481.    279.66    250.74    25474.    189.82    8152.0
0.31016E-01-0.20091E-01-0.27950E-02-0.44209E-02 0.22508E-03 0.96663E-02
```

APPENDIX C